Towards Accurate One-Stage Object Detection with AP-Loss Supplementary Material

Kean Chen[1], Jianguo Li[2], Weiyao Lin[1*], John See[3], Ji Wang[4], Lingyu Duan[5], Zhibo Chen[4], Changwei He[4], Junni Zou[1]
[1]Shanghai Jiao Tong University, China, [2] Intel Labs, China,
[3] Multimedia University, Malaysia, [4] Tencent YouTu Lab, China, [5] Peking University, China

1. Convergence

We provide proof for the proposition mentioned in Section 3.3.1 of the paper. The proof is generalized from the original convergence proof [6] for perceptron learning algorithm.

Proposition 1. *The AP-loss optimizing algorithm is guaranteed to converge in finite steps if below conditions hold:*
(1) the learning model is linear;
(2) the training data is linearly separable.

Proof. Let $\boldsymbol{\theta}$ denote the weights of the linear model. Let $\boldsymbol{f}_k^{(n)}$ denote the feature vector of k-th box in n-th training sample. Hence the score of k-th box is $s_k^{(n)} = \langle \boldsymbol{f}_k^{(n)}, \boldsymbol{\theta} \rangle$. Define $x_{ij}^{(n)} = -(s_i^{(n)} - s_j^{(n)})$. Note that the training data is separable, which means there are $\epsilon > 0$ and $\boldsymbol{\theta}^*$ that satisfy:

$$\forall n,\ \forall i \in \mathcal{P}^{(n)},\ \forall j \in \mathcal{N}^{(n)},\ \langle \boldsymbol{f}_i^{(n)}, \boldsymbol{\theta}^* \rangle \geq \langle \boldsymbol{f}_j^{(n)}, \boldsymbol{\theta}^* \rangle + \epsilon \quad (1)$$

In the t-th step, a training sample which makes an error (if there is no such training sample, the model is already optimal and algorithm will stop) is randomly chosen. Then the update of $\boldsymbol{\theta}$ is:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}(\boldsymbol{x}) \cdot (\boldsymbol{f}_i - \boldsymbol{f}_j) \quad (2)$$

where

$$L_{ij}(\boldsymbol{x}) = \frac{H(x_{ij})}{1 + \sum_{k \neq i} H(x_{ik})} \quad (3)$$

Here, since the discussion centers on the current training sample, we omit the superscript hereon.

From (2), we have

$$\begin{aligned} \langle \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^* \rangle &= \langle \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^* \rangle + \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij} \langle (\boldsymbol{f}_i - \boldsymbol{f}_j), \boldsymbol{\theta}^* \rangle \\ &\geq \langle \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^* \rangle + \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij} \epsilon \\ &\geq \langle \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^* \rangle + \max_{i \in \mathcal{P}, j \in \mathcal{N}} \{L_{ij}\} \epsilon \\ &\geq \langle \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^* \rangle + \frac{1}{|\mathcal{P}| + |\mathcal{N}|} \epsilon \end{aligned} \quad (4)$$

Hence we have

$$\langle \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^* \rangle \geq \frac{1}{|\mathcal{P}| + |\mathcal{N}|} \epsilon \cdot t \quad (5)$$

Then

$$\|\boldsymbol{\theta}^{(t)}\| \geq \frac{\langle \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^* \rangle}{\|\boldsymbol{\theta}^*\|} \geq \frac{1}{(|\mathcal{P}| + |\mathcal{N}|) \cdot \|\boldsymbol{\theta}^*\|} \epsilon \cdot t \geq c \cdot t \quad (6)$$

Here, c is a positive constant.

From (2), we also have

$$\begin{aligned} &\|\boldsymbol{\theta}^{(t+1)}\|^2 \\ =&\|\boldsymbol{\theta}^{(t)}\|^2 + \|\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}(\boldsymbol{f}_i - \boldsymbol{f}_j)\|^2 \\ &\qquad + 2\langle \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}(\boldsymbol{f}_i - \boldsymbol{f}_j), \boldsymbol{\theta}^{(t)} \rangle \\ =&\|\boldsymbol{\theta}^{(t)}\|^2 + \|\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}(\boldsymbol{f}_i - \boldsymbol{f}_j)\|^2 + 2\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij} x_{ji} \\ \leq&\|\boldsymbol{\theta}^{(t)}\|^2 + \|\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}(\boldsymbol{f}_i - \boldsymbol{f}_j)\|^2 \\ \leq&\|\boldsymbol{\theta}^{(t)}\|^2 + |\mathcal{P}| \cdot |\mathcal{N}| \cdot \max_{i \in \mathcal{P}, j \in \mathcal{N}} \{\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2\} \\ \leq&\|\boldsymbol{\theta}^{(t)}\|^2 + C \end{aligned} \quad (7)$$

Here, C is a positive constant. Hence we arrive at:

$$\|\boldsymbol{\theta}^{(t)}\|^2 \leq C \cdot t \quad (8)$$

Then, combining (6) and (8), we have

$$c^2 \cdot t^2 \leq \|\boldsymbol{\theta}^{(t)}\|^2 \leq C \cdot t \quad (9)$$

which means

$$t \leq \frac{C}{c^2} \quad (10)$$

It shows that the algorithm will stop at most after C/c^2 steps, which means that the training model will achieve the optimal solution at most after C/c^2 steps. □

*Corresponding Author, Email: wylin@sjtu.edu.cn

2. An Example of Gradient Descent Failing on Smoothed AP-loss

We approximate the step function in AP-loss by sigmoid function to make it amenable to gradient descent. Specifically, the smoothed AP-loss function is given by:

$$F = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{N}}\frac{S(x_{ij})}{1+\sum_{k\neq i}S(x_{ik})} \tag{11}$$

where

$$S(x) = \frac{e^x}{1+e^x} \tag{12}$$

Consider a linear model $s = f_1\theta_1 + f_2\theta_2$ and three training samples $(0,0),(1,0),(-3,1)$ (the first one is negative sample, others are positive samples). Then we have

$$\begin{aligned} s^{(1)} &= 0\cdot\theta_1 + 0\cdot\theta_2 \\ s^{(2)} &= 1\cdot\theta_1 + 0\cdot\theta_2 \\ s^{(3)} &= -3\cdot\theta_1 + 1\cdot\theta_2 \end{aligned} \tag{13}$$

Note that the training data is separable since we have $s^{(2)} > s^{(1)}$ and $s^{(3)} > s^{(1)}$ when $0 < \theta_1 < \frac{1}{3}\cdot\theta_2$.

Under this setting, the smoothed AP-loss become

$$\begin{aligned} F(\theta_1,\theta_2) = \frac{1}{2}\Big(&\frac{S(-\theta_1)}{1+S(-\theta_1)+S(\theta_2-4\theta_1)} \\ &+ \frac{S(3\theta_1-\theta_2)}{1+S(4\theta_1-\theta_2)+S(3\theta_1-\theta_2)}\Big) \end{aligned} \tag{14}$$

If θ_1 is sufficiently large and $\theta_1 > \theta_2 > 0$, then the partial derivatives satisfy the following condition:

$$\frac{\partial F}{\partial\theta_1} < \frac{\partial F}{\partial\theta_2} < 0 \tag{15}$$

which means θ_1 and θ_2 will keep increasing with the inequality $\theta_1 > \theta_2$ according to the gradient descent algorithm. Hence the objective function F will approach $1/6$ here. However, the objective function F approaches the global minimum value 0 if and only if $\theta_1 \to +\infty$ and $\theta_2 - 3\theta_1 \to +\infty$. This shows that the gradient descent fails to converge to global minimum in this case.

3. Inseparable Case

In this section, we will provide analysis for our algorithm with inseparable training data. We demonstrate that the bound of accumulated AP-loss depends on the best performance of learning model. The analysis is based on online learning bounds [7].

3.1. Preliminary

To handle the inseparable case, a mild modification on the proposed algorithm is needed, *i.e.* in the error-driven update scheme, L_{ij} is modified to

$$\widetilde{L}_{ij} = \frac{\widetilde{H}(x_{ij})}{1+\sum_{k\in\mathcal{P}\cup\mathcal{N},k\neq i}H(x_{ik})} \tag{16}$$

where $\widetilde{H}(\cdot)$ is defined in Section 3.4.2 (Piecewise Step Function) of the paper. The purpose is to introduce a non-zero decision margin for the pairwise score x_{ij} which makes the algorithm more robust in the inseparable case. In contrast to the case in Section 3.4.2, here we only change $H(\cdot)$ to $\widetilde{H}(\cdot)$ in the numerator for the convenience of theoretical anaysis. However, such algorithm still suffers from the discontinuity of $H(\cdot)$ in the denominator. Hence the strategy in Section 3.4.2 is also practical consideration, necessary for good performance. Then, consider the AP-loss:

$$\mathcal{L}_{AP}(\boldsymbol{x}) = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}\frac{\sum_{j\in\mathcal{N}}H(x_{ij})}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N},j\neq i}H(x_{ij})} \tag{17}$$

and define a surrogate loss function:

$$l(\boldsymbol{x},\hat{\boldsymbol{x}}) = \frac{1}{|P|}\sum_{i\in\mathcal{P}}\frac{\sum_{j\in\mathcal{N}}Q(x_{ij})}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N},j\neq i}H(\hat{x}_{ij})} \tag{18}$$

where $Q(x) = \int_{-\infty}^{x}\widetilde{H}(v)dv$. Note that the AP-loss is upper bounded by the surrogate loss:

$$l(\boldsymbol{x},\boldsymbol{x}) \geq \frac{\delta}{4}\mathcal{L}_{AP}(\boldsymbol{x}) \tag{19}$$

The learning model can be written as $\boldsymbol{x} = \boldsymbol{X}_d(\boldsymbol{\theta})$, where $d \in \mathcal{D}$ denotes the training data for one iteration and D is the whole training set. Then, the modified error-driven algorithm is equivalent to gradient descent on surrogate loss $l(\boldsymbol{X}_{d^{(t)}}(\boldsymbol{\theta}),\boldsymbol{X}_{d^{(t)}}(\boldsymbol{\theta}^{(t)}))$ at each step t. We further suppose below conditions are satisfied:

(1) For all $\hat{\boldsymbol{\theta}}$ and $d\in\mathcal{D}$, $l(\boldsymbol{X}_d(\boldsymbol{\theta}),\boldsymbol{X}_d(\hat{\boldsymbol{\theta}}))$ is convex *w.r.t* $\boldsymbol{\theta}$.

(2) For all $d\in\mathcal{D}$, $\|\partial\boldsymbol{X}_d(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\|$ is upper bounded by a constant R. Here $\|\cdot\|$ is the matrix norm induced by the 2-norm for vectors.

Remark 1. Note that these two conditions are satisfied if the learning model is linear.

3.2. Bound of Accumulated Loss

By the convexity, we have:

$$l^{(t)}(\boldsymbol{\theta}) \leq l^{(t)}(\boldsymbol{u}) + \langle\boldsymbol{\theta}-\boldsymbol{u},\frac{\partial l^{(t)}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\rangle \tag{20}$$

where we use $l^{(t)}(\boldsymbol{\theta})$ to denote $l(\boldsymbol{X}_{d^{(t)}}(\boldsymbol{\theta}),\boldsymbol{X}_{d^{(t)}}(\boldsymbol{\theta}^{(t)}))$ and $\boldsymbol{u}$ can be any vector of model weights. Then, let $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$ and compute the sum over $t = 1\sim T$, we have:

$$\begin{aligned} &\sum_{t=1}^{T}l^{(t)}(\boldsymbol{\theta}^{(t)}) - \sum_{t=1}^{T}l^{(t)}(\boldsymbol{u}) \leq \sum_{t=1}^{T}\langle\boldsymbol{\theta}^{(t)}-\boldsymbol{u},\frac{\partial l^{(t)}(\boldsymbol{\theta}^{(t)})}{\partial\boldsymbol{\theta}}\rangle \\ &= \sum_{t=1}^{T}\langle\boldsymbol{\theta}^{(t)}-\boldsymbol{u},\frac{1}{\eta}(\boldsymbol{\theta}^{(t)}-\boldsymbol{\theta}^{(t+1)})\rangle \\ &\leq \frac{1}{2\eta}\|\boldsymbol{u}-\boldsymbol{\theta}^{(1)}\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T}\|\boldsymbol{\theta}^{(t)}-\boldsymbol{\theta}^{(t+1)}\|^2 \\ &= \frac{1}{2\eta}\|\boldsymbol{u}-\boldsymbol{\theta}^{(1)}\|^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\frac{\partial l^{(t)}(\boldsymbol{\theta}^{(t)})}{\partial\boldsymbol{\theta}}\|^2 \end{aligned} \tag{21}$$

where η is the step size of gradient descent. Note that

$$\frac{\partial l^{(t)}(\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{X}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t)}} \cdot \frac{\partial l(\boldsymbol{x}, \boldsymbol{x}^{(t)})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\boldsymbol{X}(\boldsymbol{\theta}^{(t)})} \tag{22}$$

and

$$\begin{aligned}\|\frac{\partial l(\boldsymbol{x}, \boldsymbol{x}^{(t)})}{\partial \boldsymbol{x}}\|^2 &= \frac{1}{|\mathcal{P}|^2}\sum_{i\in\mathcal{P}} \frac{\sum_{j\in\mathcal{N}} \widetilde{H}^2(x_{ij})}{(1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(x_{ij}^{(t)}))^2} \\ &\le \frac{1}{|\mathcal{P}|^2}\sum_{i\in\mathcal{P}} \frac{\frac{1}{\delta}\sum_{j\in\mathcal{N}} Q(x_{ij})}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(x_{ij}^{(t)})} \le \frac{1}{\delta} l(\boldsymbol{x}, \boldsymbol{x}^{(t)})\end{aligned} \tag{23}$$

Note that both $\mathcal{P}_d$ and $\mathcal{N}_d$ depend on d. However, we omit the subscript d here since the discussion only centers on the current training sample $d^{(t)}$.

Hence we have:

$$\begin{aligned}&\sum_{t=1}^{T} l^{(t)}(\boldsymbol{\theta}^{(t)}) - \sum_{t=1}^{T} l^{(t)}(\boldsymbol{u}) \\ &\le \frac{1}{2\eta}\|\boldsymbol{u} - \boldsymbol{\theta}^{(1)}\|^2 + \frac{\eta R^2}{2\delta}\sum_{t=1}^{T} l^{(t)}(\boldsymbol{\theta}^{(t)}).\end{aligned} \tag{24}$$

Let $\eta = \delta/R^2$, rearrange and get the expression:

$$\frac{1}{2}\sum_{t=1}^{T} l^{(t)}(\boldsymbol{\theta}^{(t)}) \le \frac{R^2}{2\delta}\|\boldsymbol{u} - \boldsymbol{\theta}^{(1)}\|^2 + \sum_{t=1}^{T} l^{(t)}(\boldsymbol{u}) \tag{25}$$

This entails the bound of surrogate loss l:

$$\sum_{t=1}^{T} l^{(t)}(\boldsymbol{\theta}^{(t)}) \le 2\sum_{t=1}^{T} l^{(t)}(\boldsymbol{u}) + \frac{R^2}{\delta}\|\boldsymbol{u} - \boldsymbol{\theta}^{(1)}\|^2 \tag{26}$$

which implies the bound of AP-loss $\mathcal{L}_{AP}$:

$$\sum_{t=1}^{T} \mathcal{L}_{AP}(\boldsymbol{X}(\boldsymbol{\theta}^{(t)})) \le \frac{8}{\delta}\sum_{t=1}^{T} l^{(t)}(\boldsymbol{u}) + \frac{4R^2}{\delta^2}\|\boldsymbol{u} - \boldsymbol{\theta}^{(1)}\|^2 \tag{27}$$

As a special case, if there exists a $\boldsymbol{u}$ such that $l^{(t)}(\boldsymbol{u}) = 0$ for all t, then the accumulated AP-loss is bounded by a constant, which implies that convergence can be achieved with finite steps (similar to that of the separable case). Otherwise, with sufficiently large T, the average AP-loss mainly depends on $\frac{1}{T}\frac{8}{\delta}\sum_{t=1}^{T} l^{(t)}(\boldsymbol{u})$. This implies that the bound is meaningful if there still exists a sufficiently good solution $\boldsymbol{u}$ in such inseparable case.

3.3. Offline Setting

With the offline setting ($d^{(t)} = d$ for all t), a bound with simpler form can be revealed. For simplicity, we will omit the subscript d of $\boldsymbol{X}_d(\boldsymbol{u}), \mathcal{P}_d, \mathcal{N}_d$ and define $A_i(\boldsymbol{u}) = \sum_{j\in\mathcal{N}} Q(X_{ij}(\boldsymbol{u}))$, $Z(\boldsymbol{u}) = \max_{i\in\mathcal{P}}\{A_i(\boldsymbol{u})\}$. Then,

$$\begin{aligned}l^{(t)}(\boldsymbol{u}) &= \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \frac{\sum_{j\in\mathcal{N}} Q(X_{ij}(\boldsymbol{u}))}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))} \\ &= \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \frac{A_i(\boldsymbol{u})}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))} \\ &\le \frac{Z(\boldsymbol{u})}{|\mathcal{P}|}\sum_{i=1}^{|\mathcal{P}|}\frac{1}{i} \le \frac{\ln|\mathcal{P}|+1}{|\mathcal{P}|} Z(\boldsymbol{u})\end{aligned} \tag{28}$$

The second last inequality is based on the fact that $(1 + \sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)})))$ are picked from $1 \sim (|\mathcal{P}| + |\mathcal{N}|)$ without replacement (assume no ties; if ties exist, this inequality still holds). Combining the results from Equation 28 and Equation 27, we have:

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{AP}(\boldsymbol{X}(\boldsymbol{\theta}^{(t)})) \le \frac{\ln|\mathcal{P}|+1}{|\mathcal{P}|}\cdot\frac{8}{\delta}Z(\boldsymbol{u}) + \frac{1}{T}\frac{4R^2\|\boldsymbol{u} - \boldsymbol{\theta}^{(1)}\|^2}{\delta^2} \tag{29}$$

Next,

$$\begin{aligned}l^{(t)}(\boldsymbol{u}) &= \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \frac{A_i(\boldsymbol{u})}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))} \\ &= \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \frac{1+\sum_{j\in\mathcal{P}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))} \\ &\qquad \cdot \frac{A_i(\boldsymbol{u})}{1+\sum_{j\in\mathcal{P}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))} \\ &\le \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \frac{1+\sum_{j\in\mathcal{P}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))}{1+\sum_{j\in\mathcal{P}\cup\mathcal{N}, j\neq i} H(X_{ij}(\boldsymbol{\theta}^{(t)}))}\cdot Z(\boldsymbol{u}) \\ &= (1 - \mathcal{L}_{AP}(\boldsymbol{X}(\boldsymbol{\theta}^{(t)})))\cdot Z(\boldsymbol{u})\end{aligned} \tag{30}$$

Combining the results from Equation 30 and Equation 27, we have:

$$\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{AP}(\boldsymbol{X}(\boldsymbol{\theta}^{(t)})) \le \frac{\frac{8}{\delta}Z(\boldsymbol{u})}{1+\frac{8}{\delta}Z(\boldsymbol{u})} + \frac{1}{T}\frac{4R^2\|\boldsymbol{u} - \boldsymbol{\theta}^{(1)}\|^2}{\delta^2} \tag{31}$$

If $Z(\boldsymbol{u})$ is small, the bound in Equation 29 is active, otherwise the bound in Equation 31 is active. Consequently, we have:

$$\overline{\mathcal{L}_{AP}} \le \min\{\frac{\ln|\mathcal{P}|+1}{|\mathcal{P}|}\frac{8}{\delta}Z(\boldsymbol{u}), \frac{\frac{8}{\delta}Z(\boldsymbol{u})}{1+\frac{8}{\delta}Z(\boldsymbol{u})}\} + \epsilon \tag{32}$$

where $\overline{\mathcal{L}_{AP}}$ denotes the average AP-loss, $\epsilon \to 0$ as T increases.

4. Consistency

Observation 1. *When the activation function $L(\cdot)$ takes the form of softmax function and loss-augmented step function, our optimization algorithm can be expressed as the gradient descent algorithm on cross-entropy loss and hinge loss respectively.*

Cross Entropy Loss: Consider the multi-class classification task. The outputs of neural network are $(x_1, \ldots, x_K)$

where K is the number of classes, and the ground truth label is $y \in \{1, \ldots, K\}$. Using softmax as the activation function, we have:

$$\begin{aligned}&(L_1, \ldots, L_K) \\ &= softmax(\boldsymbol{x}) = (\frac{e^{x_1}}{\sum_i e^{x_i}}, \ldots, \frac{e^{x_K}}{\sum_i e^{x_i}})\end{aligned} \tag{33}$$

The cross entropy loss is:

$$\mathcal{L}_{ce} = -\sum_i \mathbf{1}_{y=i} \log(L_i) \tag{34}$$

Hence the gradient of x_i is

$$g_i = L_i - \mathbf{1}_{y=i} \tag{35}$$

Note that g_i is "error-driven" with the desired output $\mathbf{1}_{y=i}$ and current output L_i. This form is consistent with our error-driven update scheme (c.f. Section 3.2.1 of the paper).

Hinge Loss: Consider the binary classification task. The output of neural network is x, and the ground truth label is $y \in \{1, 2\}$. Define $(x_1, x_2) = (-x, x)$. Using loss-augmented step function as the activation function, we have:

$$(L_1, L_2) = (H(x_1 - 1), H(x_2 - 1)) \tag{36}$$

where $H(\cdot)$ is the Heaviside step function. The hinge loss is:

$$\mathcal{L}_{hinge} = \mathbf{1}_{y=1} \max\{1 - x_1, 0\} + \mathbf{1}_{y=2} \max\{1 - x_2, 0\} \tag{37}$$

Hence the gradient of x_i is

$$g_i = \mathbf{1}_{y=i} \cdot (L_i - 1) \tag{38}$$

There are two cases. If $y = i$, the gradient g_i is "error-driven" with the desired output 1 and current output L_i. If $y \neq i$, the gradient g_i equals zero, since x_i does not contribute to the loss. This form is consistent with our error-driven update scheme (c.f. Section 3.2.1 of the paper).

5. Additional Experiments on SSD

5.1. Experimental Settings

We also evaluate the proposed AP-loss on another one-stage detector SSD [5]. The models are trained on VOC2007 and VOC2012 `trainval` sets, and tested on VOC2007 `test` set. We use VGG-16 [8] as the backbone model which is pre-trained on the ImageNet-1k classification dataset [1]. We use `conv4_3`, `conv7`, `conv8_2`, `conv9_2`, `conv10_2`, `conv11_2`, `conv12_2` to predict both location and their corresponding confidences. An additional convolution layer is added after `conv4_3` to scale the feature. The associated anchors are the same as that designed in [5]. In testing phase, the input image is fixed to 512×512. For focal loss with SSD, we observe that the hyper-parameters $\gamma = 1, \alpha = 0.25$ lead to a much better performance than the original settings in [3] which are $\gamma = 2, \alpha = 0.25$. Hence we evaluate the focal loss with new γ and α in our experiments on SSD. Other details are similar to that in Section 4.1 of the paper.

Training Loss	PASCAL VOC		
	AP	AP_{50}	AP_{75}
CE-Loss + OHEM	43.6	76.0	44.7
Focal Loss	39.3	69.9	38.0
AUC-Loss	33.8	63.7	31.5
AP-Loss	**45.2**	**77.3**	**47.3**

Table 1: Comparison through different training losses. Models are tested on VOC2007 `test` set. The metric AP is averaged over multiple IoU thresholds of $0.50 : 0.05 : 0.95$.

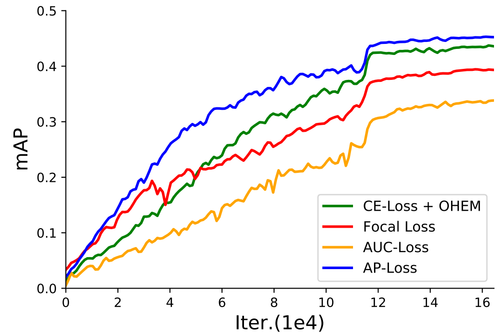

Figure 1: Detection accuracy (mAP) on VOC2007 `test` set.(Best viewed in color)

5.2. Results

The results are shown in Table 1 and Figure 1. Note that the AP-loss outperforms all the other losses at both the final state and various snapshot time points. Together with the results on RetinaNet [3] in Section 4.2.2 of the paper, we observe the robustness of the proposed AP-loss, which performs much better than the other competing losses on different datasets (*i.e.* PASCAL VOC [2], MS COCO [4]) and different detectors (*i.e.* RetinaNet [3], SSD [5]). This demonstrates the effectiveness and strong generalization ability of our proposed approach.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[2] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[3] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Trans on PAMI*, 2018.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[6] A Novikoff. On convergence proofs for perceptrons. *Proc.sympos.math.theory of Automata*, pages 615–622, 1963.

[7] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.